

Syncro: Universal Reverse-Auction Marketplace

Project Workflow and Timeline

Team CodeBuds

E/22/052, E/22/419, E/22/058, E/22/121

January 12, 2026

1. Introduction

Syncro is a dual-role marketplace designed to support two distinct transaction methods which are direct procurement and a reverse-auction bidding system. The project allows a single account to function as both a Client and a Seller. This report provides a detailed 29-week plan to build the platform. We have divided the work into two main parts. They are,

Phase 1: Search and Direct Orders

Phase 2: Bidding Engine and AI Integration.

2. Development Approach

We are using an Incremental Development strategy, which allows us to build and refine the platform in stages. To ensure steady progress and meet our deadlines, we have established a detailed weekly plan that tracks specific technical tasks for both the frontend and backend. This approach enables us to launch a stable version of the core marketplace by April 13 while continuing to develop complex features like the AI-powered consultant and real-time bidding for the final July deadline.

3. Project Timeline and Workflow

Phase 1: Direct Orders, Search, and Communication

Goal: Build the core marketplace foundation. By April 13, buyers should be able to find sellers and place direct orders.

January: System Design & Foundation

- **Week 1 (Jan 12–18): Architecture & Requirement Analysis**
 - **Finalize ERD:** Design a relational database schema that handles "Dual-Role" users and "Service-to-Category" mappings.
 - **Tech Stack Initialization:** Set up the FastAPI (Python) backend and Next.js (React) frontend repositories.
 - **Workflow Definition:** Establish a GitHub project board with a "To-Do, In-Progress, Done" Kanban structure.
- **Week 2 (Jan 19–25): Database & Environment Setup**
 - **PostgreSQL Setup:** Initialize the database and create tables for Users, Profiles, and Categories.
 - **Pydantic Schemas:** Write data validation schemas in FastAPI to ensure secure data transfer.
 - **CRUD Boilerplate:** Develop basic Create, Read, Update, and Delete endpoints for the listing system.
- **Week 3 (Jan 26 – Feb 1): Authentication & User Profiles**
 - **JWT Implementation:** Set up JSON Web Token (JWT) authentication for secure login.
 - **Role Logic:** Code the logic that allows one account to toggle between a "Client" view and a "Seller" dashboard.
 - **Frontend Shell:** Build the landing page and the initial login/signup UI components.

February: Discovery & Listing Features

- **Week 4 (Feb 2–8): Seller Dashboard & Storefront**
 - **Business Profile UI:** Build the form for sellers to add their business name, bio, and niche specialties.
 - **Portfolio Integration:** Develop the backend logic to link specific listings (products/services) to a unique Seller ID.
- **Week 5 (Feb 9–15): Media Management & Listings**
 - **Cloudinary Integration:** Connect the backend to Cloudinary for cloud-based image hosting of product photos.
 - **Listing Form:** Create the "Service Listing" form with fields for Title, Description, and Fixed Price.
- **Week 6 (Feb 16–22): Mid-Semester Break (Internal Review)**
 - **Code Audit:** Conduct a peer-review session to fix any technical debt in the authentication or listing modules.
 - **Documentation:** Write the initial API documentation using FastAPI's automatic Swagger UI.
- **Week 7 (Feb 23 – Mar 1): Advanced Search & Filtering**
 - **SQL Search Queries:** Implement PostgreSQL full-text search to let buyers find sellers by keyword.
 - **Category Filters:** Build the frontend sidebar to filter results by Industry (Catering, Tutoring, etc.).

March: Transactional Logic & Real-time Chat

- **Week 8 (Mar 2–8): Direct Ordering Workflow**
 - **Order Creation:** Build the "Order" button that generates a record in the database with a "Pending" status.
 - **Client Dashboard:** Design a page where clients can track the status of their active orders.
- **Week 9 (Mar 9–15): Communication Engine (Socket.io)**
 - **WebSocket Setup:** Initialize a real-time communication server using python-socketio.
 - **Message Persistence:** Create a database table to store chat history between buyers and sellers.
- **Week 10 (Mar 16–22): Negotiation & Custom Offers**
 - **Price Adjustment:** Develop a feature where sellers can send a "Negotiated Price" offer within the chat.
 - **Client Verification:** Build the frontend button for clients to "Accept Offer," which updates the final order price in the database.
- **Week 11 (Mar 23–29): Frontend Polish & Integration**
 - **State Management:** Use React state management (like Redux or Context API) to handle live updates across the app.
 - **UX Design:** Add "Loading" states, "Success" toasts, and professional transitions.
- **Week 12 (Mar 30 – Apr 5): Phase 1 Quality Assurance**
 - **End-to-End Testing:** Simulate the entire process from searching a seller to placing and negotiating an order.
 - **Bug Squashing:** Fix any UI layout issues or backend response errors before the milestone.

Phase 2: Syncro RFP, Bidding, and AI Intelligence

Goal: Implement the "Reverse Auction" model where sellers compete for client requests.

April: The RFP Transition

- **Week 13 (Apr 6–12): Phase 1 Milestone Release (April 13)**
 - **Deployment:** Launch the "Search & Direct Order" version of Syncro.
 - **Monitoring:** Monitor for any post-launch bugs from real user interactions.
- **Week 14 (Apr 13–19): New Year Break (Design Review)**
 - **Phase 2 Scoping:** Finalize the "RFP" (Request for Proposal) data requirements for the bidding phase.
- **Week 15 (Apr 20–26): RFP Posting System**
 - **Dynamic RFP Form:** Build the "Post a Request" interface for clients to list their specific needs (quantity, budget, deadline).
 - **Categorization:** Ensure every RFP is tagged with a niche so the system knows which sellers to alert.
- **Week 16 (Apr 27 – May 3): Seller Lead Feed**
 - **Bidding Board:** Develop the dashboard where sellers view a feed of active requests relevant to their niche.

May: Bidding Logic & Smart Notifications

- **Week 17 (May 4–10): The Bidding Engine**
 - **Bid Creation:** Build the backend logic for sellers to submit "Counter-Offers" or "Bids" on a client's RFP.
 - **Price Validation:** Ensure bids stay within a logical range relative to the client's budget.
- **Week 18 (May 11–17): Real-time Matching & Alerts**
 - **Niche Notifications:** Implement an automated alert system that emails or pushes notifications to sellers when a matching RFP is posted.
- **Week 19 (May 18–24): Bid Comparison UI**
 - **Comparison Dashboard:** Build a table for clients to compare all received bids based on Price, Delivery Date, and Seller Rating.
- **Week 20 (May 25–31): Live Bidding Updates**
 - **Dynamic Auction Feel:** Use WebSockets to show new bids appearing in real-time on the client's screen.

June: AI Integration & Platform Management

- **Week 21 (June 1–7): AI Integration (Gemini/OpenAI)**
 - **Intelligent Elicitation:** Setup the API connection to assist clients in writing professional RFPs.
- **Week 22 (June 8–14): Chatbot Customization**
 - **Contextual Interviewing:** Train the bot to ask follow-up questions (e.g., asking about guest count for catering requests).

- **Week 23 (June 15–21): Commission & Monetization Engine**
 - **Platform Fee Logic:** Code the backend to automatically calculate the 5-10% fee from the seller's final bid.
- **Week 24 (June 22–28): Security Audit**
 - **Data Protection:** Review the security of the commission engine and private chat logs.

July: Testing & Handover

- **Week 25 (June 29 – July 5): Full Integration Testing**
 - **System Check:** Ensure a user can switch roles and use both Direct Ordering and the Bidding system seamlessly.
- **Week 26 (July 6–12): Automated & Stress Testing**
 - **Performance Check:** Test how the bidding system handles multiple simultaneous bids on a single request.
- **Week 27 (July 13–19): User Acceptance Testing (UAT)**
 - **Feedback Loop:** Have peer groups test the system and identify UX friction points.
- **Week 28 (July 20–26): Final Documentation & Polishing**
 - **Report Finalization:** Finalize technical manuals, API docs, and project presentation.
- **Week 29 (July 27–31): Final Submission & Presentation**
 - **Project Handover:** Successful deployment and final assessment presentation.

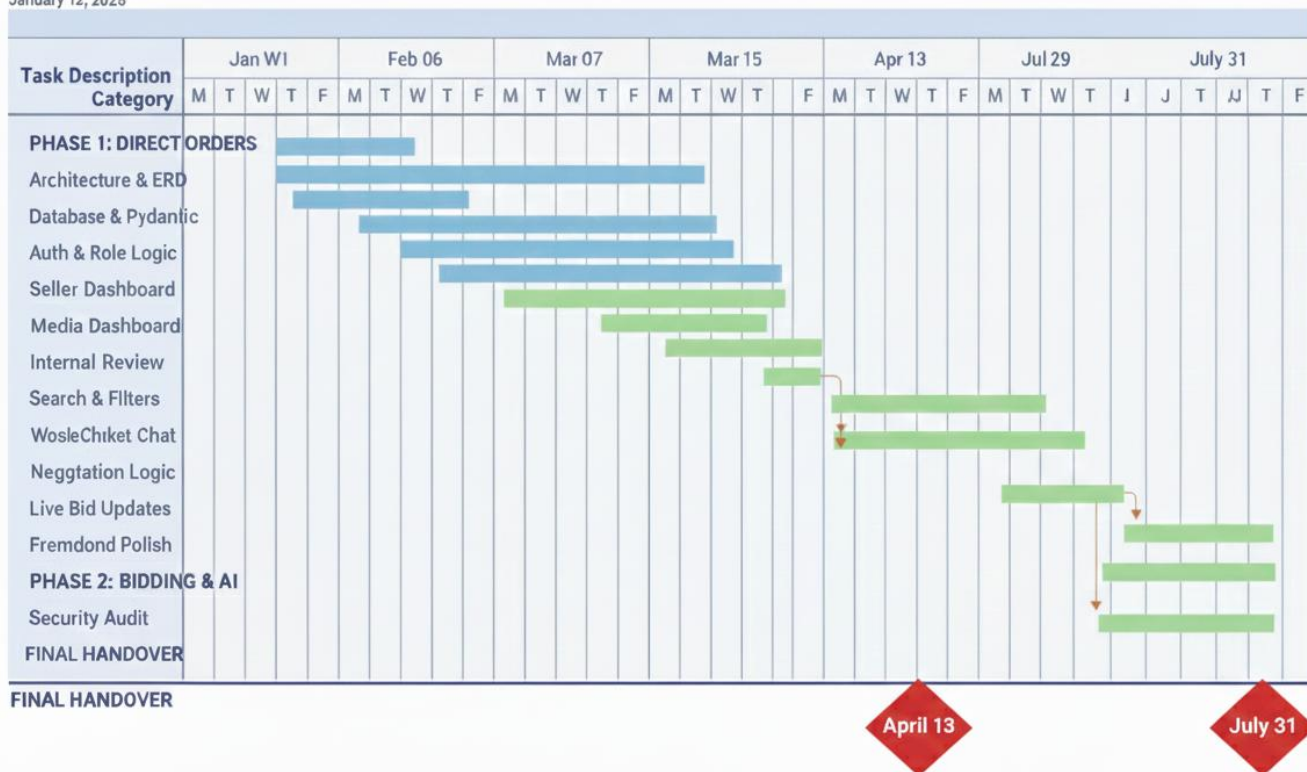


Figure 1: Weekly development timeline for Syncro (Phase 1 and Phase 2)